# House Price Prediction

Akshara Narayana, Prashanth Adapa, Ghanashri Mariyanna, Pratik Joseph Dabre
Department of Software Engineering, San Jose State University
San Jose, California.
akshara.narayana@sjsu.edu, venkatapavanprashanth.adapa@sjsu.edu, ghanashri.mariyanna@sjsu.edu, pratikjoseph.dabre@sjsu.edu

***Abstract: To create a machine learning model capable of predicting house prices based on various describing attributes. The dataset used is the Ames Housing dataset and it has 79 explanatory variables that describe every aspect of Ames, Iowa's residential homes. The data is cleaned first, and then imputed into various machine learning models for performance comparison. The data analysis and observations summarized in this paper are used to finalize and use a machine learning model that can effectively predict housing prices, with the understanding that the algorithm can still be improved using advanced machine learning algorithms.***

## I. INTRODUCTION

This project focuses on building an effective machine learning model that can accurately predict house prices in Ames, Iowa. With property and real estate prices rising not only in the United States but around the world, it is critical to conduct this analysis. According to our research, house prices cannot be solely determined by a few factors, but rather by a combination of factors that affect the overall price of the house. Furthermore, house prices are almost never effectively predicted using only a few variables. Houses have a variety of features that may not be the same price due to their location. For example, a large house may be worth more if it is in a desirable rich neighborhood rather than in a poor neighborhood. We worked hard to identify the important factors that influence house prices in Iowa and to develop a model that can reasonably predict house prices.

This dataset consists of 79 features that describe the features of a house in detail. There are approximately equal number of categorical and numerical attributes that have varying relationship with the Target variable Sale Price. To improve prediction accuracy, the data used in the experiment will be handled using a combination of pre-processing methods. Furthermore, some variables will be added to the local dataset to investigate the relationship between these variables and the sale price in Ames.

*The Features are: Order, PID, MS SubClass, MS Zoning, Lot Frontage, Lot Area, Street, Alley, Lot Shape, Land Contour,Utilities ,Lot Config, Land Slope,Neighborhood, Condition 1, Condition 2, Bldg Type, House Style, Overall Qual, Overall Cond, Year Built, Year Remod/Add, Roof Styl , Roof Matl , Exterior 1st,Exterior 2nd, Mas Vnr Type , Mas Vnr Area, Exter Qual, Exter Cond, Foundation, Bsmt Qual, Bsmt Cond,Bsmt Exposure, BsmtFin Type 1, BsmtFin SF 1, BsmtFin Type 2, BsmtFin SF 2 ,Bsmt Unf SF,Total Bsmt SF,Heating,Heating QC ,Central Air ,Electrical,1st Flr SF, 2nd Flr SF,Low Qual Fin SF ,Gr Liv Area, Bsmt Full Bath,Bsmt Half Bath, Full Bath,Half Bath ,Bedroom AbvGr, Kitchen AbvGr ,Kitchen Qual ,TotRms AbvGrd, Functional Fireplaces ,Fireplace Qu, Garage Type, Garage Yr Blt ,Garage Finish, Garage Cars, Garage Area ,Garage Qual, Garage Cond, Paved Drive, Wood Deck SF,Open Porch SF, Enclosed Porch,3Ssn Porch, Screen Porch, Pool Area, Pool QC, Fence, Misc Feature, Misc Val ,Mo Sold ,Yr Sold, Sale Type, Sale Condition, SalePrice.*

To find the best performing model for this dataset, various types of machine learning models were used and compared. We were able to identify a model that could identify house prices with reasonable accuracy.

## II. DATA ANALYSIS

### A. DATASET

This project uses the Ames housing data available on Kaggle, which includes 81 features describing a wide range of characteristics of 1,459 homes in Ames, Iowa sold between 2010 and 2019.

### B. EXPLORATORY DATA ANALYSIS

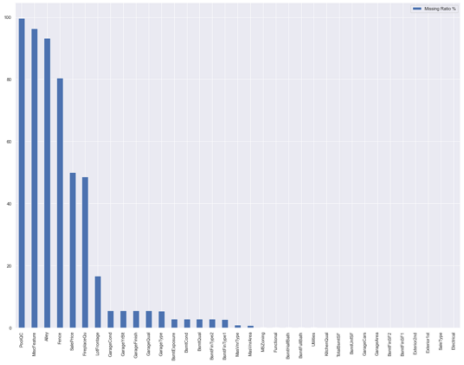
#### 1. Data Pre-Processing

It is a data mining approach for converting raw data into a usable and efficient format to derive meaningful information out of it.

a. **Finding Null Values**: Initial EDA showed that several features had missing values, which would need to be handled appropriately before modelling. We plotted a bar graph that gave us an estimation on the number of null values in each column. We also plotted a dendrogram plot provides a tree-like graph generated through hierarchical clustering and groups together columns that have strong correlations in nullity. Method used: msno.dendrogram

b. **Data Cleaning**: There may be various useless and missing elements in the raw data. Data cleaning is used to deal with this aspect. It entails dealing with missing data and noisy data. Method used: plot_nas()

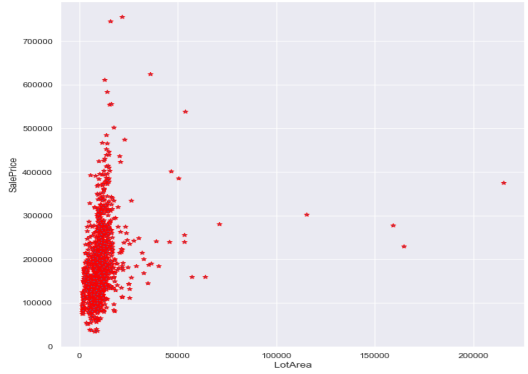
Fig 1: Plot showing percentage of null values



**Observation:** We calculated the percentage of missing values in each feature and observed that there where four features with null values greater than 80%, therefore we reduced shape of the dataframe by dropping these columns.

c. **Outlier Detection:** Outliers are occurrences or observations which appears to be inconsistent with the remainder of that set of data. Outliers are cases that are unusual because they fall outside the distribution that is considered normal for the data. The distance from the center of a normal distribution indicates how typical a given point is with respect to the distribution of the data. The presence of outliers can have a deleterious effect on many forms of data mining. Method used: outlier_visualization()

Fig 2: Plot showing outliers in the data.

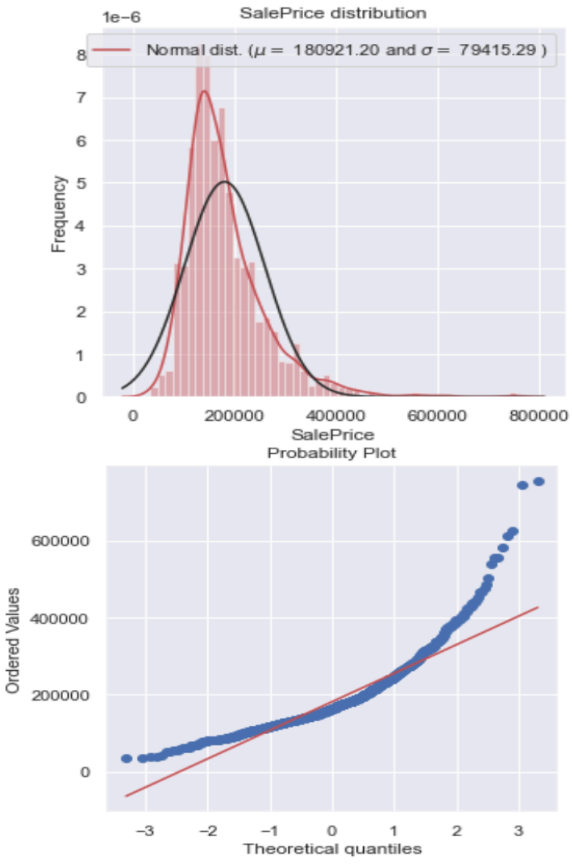



**Observation**: Considering the importance of property size metrics, we preferred to observe the outliers of two important features "LotArea" and "GrLivArea". From the scatter plots, it was observed that few values are abnormal and would not be useful to our models. Hence, defined a range for LotArea, GrLivArea with respect to Sale Price and removed the values that were falling out of the defined range.

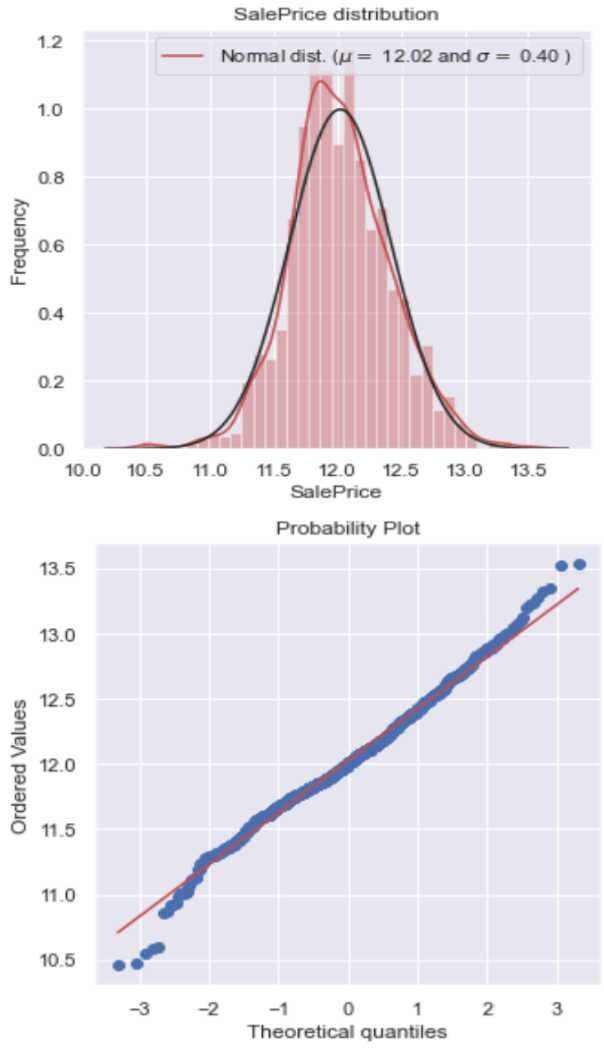
## 2. Feature Engineering

a. As part of feature engineering, we plotted the distribution plots and observed that it is right skewed as shown below. After extensive research, we decided to use the log transformation method to handle the skewness. Method used: Log Transformation

Fig 3: Distribution plots showing skewness.



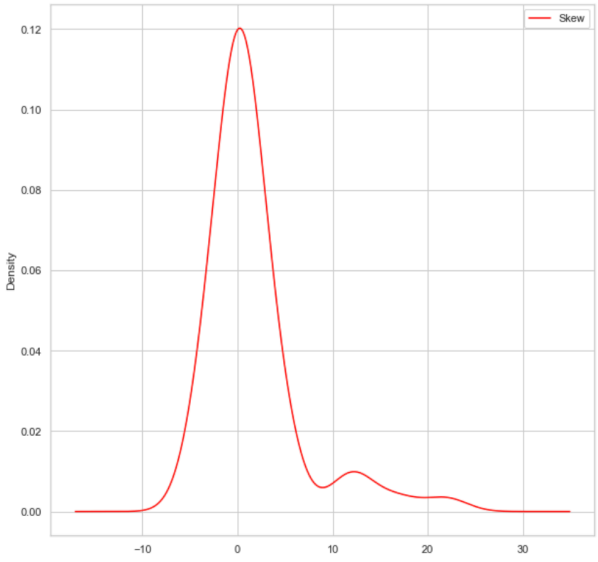

**Observation**: We implemented the Log Transformation of skewed target variable – SalePrice. Log-transformation is a technique used to perform Feature Transformation. It is one of the many techniques that can be used to transform the features so that they are treated equally. This method helps to handle skewed data and after transformation, the distribution becomes more approximate to normal. Log-Tranform method is majorly used to decreases the effect of the outliers, due to the normalization of magnitude differences so that the model becomes more robust. After the log transformation, we again plotted a graph with respect to the quantiles of our target feature against the quantiles of a normal distribution.



- Feature Addition: After extensive analysis and observation of the relationships between the features, we found that combining few features and generating new feature would be a positive addition to our analysis and models. Hence, we generated two features, ‘Age_of_house’ and ‘TotalSF’.

b. We again checked for skewness of the features and observed that few of the features still have some skew, so performed Box Cox Transformation of highly skewed variables to handle this situation.

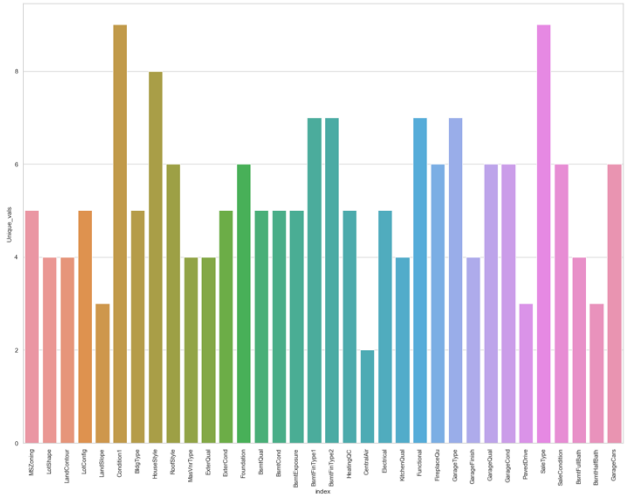
Fig 4: plot showing skewness




## 3. Data Visualizations

Data visualization is the representation of data through use of common graphics, such as charts, plots, infographics, and even animations. These visual displays of information communicate complex data relationships and data-driven insights in a way that is easy to understand.

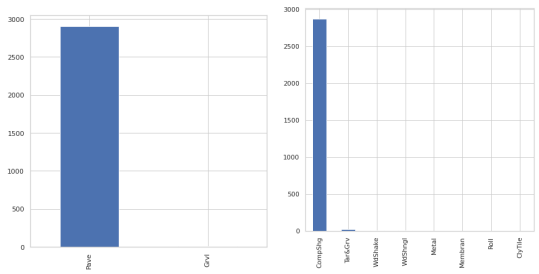
i. Number of unique values in each feature: It is observed that there are only 33 features with univque values less than 9.



ii. Column wise variance plotting

Whenever there are columns in a data frame with only one distinct value, those columns will have zero variance. In fact, the reverse is true too; a zero-variance column will always have exactly one distinct value. The proof of the former statement follows directly from the definition of variance. The proof of the reverse, however, is based on measure theory - specifically that if the expectation of a non-negative random variable is zero then the random variable is equal to zero.
The existance of zero variance columns in a data frame seemed benign in predicting house prices. We performed variance plotting for all categorical columns to indentify any uneven distribution of data.
Constant features show similar/single values in all the observations in the dataset. We concluded the features which provide no information that allows ML models to predict the target and dropped them from our dataset.

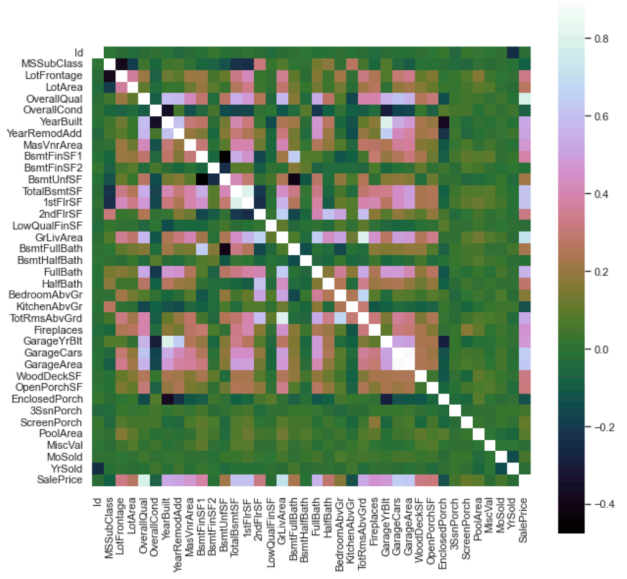
Fig 5: Plot showing percentage variance



### 4. Correlation Analysis

Correlation analysis calculates the extent to which one variable change due to changes in other variables. A high correlation indicates a strong association between the two variables, and a low correlation indicates a weak association between the variables. Correlation can be used to test hypotheses about cause-and-effect relationships between variables. To better understand these patterns and relationships, we have plotted a heatmap.

Fig 6: correlation heatmap




Observation: We have observed that there are ten features that are strongly correlated with our target variable, they are: *OverallQual, YearBuilt, YearRemodAdd, TotalBsmtSF, 1stFlrSF, GrLivArea, FullBath, TotRmsAbvGrd, GarageCars, GarageArea.*

| | corr value |
|---|---|
| **OverallQual** | 0.790982 |
| **YearBuilt** | 0.522897 |
| **YearRemodAdd** | 0.507101 |
| **TotalBsmtSF** | 0.613581 |
| **1stFlrSF** | 0.605852 |
| **GrLivArea** | 0.708624 |
| **FullBath** | 0.560664 |
| **TotRmsAbvGrd** | 0.533723 |
| **GarageCars** | 0.640409 |
| **GarageArea** | 0.623431 |
| **SalePrice** | 1.000000 |

Fig 7: correlation score

## III. METHODS

### 1. MULTIPLE LINEAR REGRESSION

Multiple linear regression (MLR), also known simply as multiple regression, is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of multiple linear regression is to model the linear relationship between the explanatory (independent) variables and response (dependent) variables. In essence, multiple regression is the extension of ordinary least-squares (OLS) regression because it involves more than one explanatory variable.

Even though a linear model may be optimal for the data given to create the model, it is not necessarily guaranteed to be the best model for predictions on unseen data. Our underlying data followed a relatively simple model, and the model we use is too complex for the task and what we are essentially doing is that we are putting too much weight on any possible change or variance in the data. Our model is overreacting and overcompensating for even the slightest change in our data. We have features in our dataset that are linearly correlated with other features.

### 2. GRADIENT BOOSTING

Gradient Boosting Algorithm is generally used when we want to decrease the Bias error. The gradient boosting regression model was chosen as the second model because it is an ensemble

method that creates multiple weak models and then combines them to improve performance. We chose this model because linear regression aims to draw a line that perfectly fits your data. Gradient boosting attempts to improve this by first selecting a very simple solution and then attempting to improve the model based on the results/errors of previous iterations. The results of gradient descent algorithms revealed that the model was overfitting, which means it could only predict seen data.

## 3. XGBOOST REGRESSOR

The third model chosen was xgboost regressor because it is a more efficient and effective implementation of gradient boosting. This regression model has the reputation to improve model performance and to mitigate some of the issues encountered in gradient boosting regression. Surprisingly, the model was unable to alleviate any of the problems encountered by the gradient boosting regression. The model took too much time to run and the results were disappointing. We concluded that the reason we are experiencing inefficiencies in our models is due to model overfitting.

**Solution to avoid overfitting**: There are Regression techniques to avoid overfitting by adding a penalty to models that have too large coefficients. This means that the estimated coefficients are pushed towards 0, to make them work better on new datasets ("optimized for prediction"). This allows you to use complex models and avoid over-fitting at the same time. In short, ridge regression and lasso are regression techniques optimized for prediction, rather than inference. Ridge and lasso regression allow you to regularize ("shrink") coefficients.

## 4. RIDGE REGRESSION

Ridge regularization handled the model complexity by focusing more on the important features which contributed more to the overall error than the less important features. But still, it used information from less important features in the model. Different features contributed differently to the overall error and naturally our quest is to focus more on the important features which contribute more to the error than less important ones which can be handled by the Ridge regularization.

## 5. LASSO REGRESSION

Since we have a high dimensionality and high correlation in our dataset, we preferred to try Lasso regularization since it penalizes less important features more and makes them zero which gives us the benefit of algorithmic feature selection and would make robust predictions than Ridge regularization but sometimes it can remove certain signals from the model even when they have information so it should be used carefully. To conclude, we have used this model because our dataset displayed high multicollinearity and we tried to automate variable elimination and feature selection.

## 6. RIDGE AND LASSO WITH HYPERPARAMETERS

Using the terminology from “The Elements of Statistical Learning,” a hyperparameter “alpha” is provided to assign how much weight is given to each of the L1(Ridge) and L2(Lasso) penalties. Alpha is a value between 0 and 1 and is used to weight the contribution of the L1 penalty and one minus the alpha value is used to weight the L2 penalty. Alpha is a value between 0 and 1 and is used to weight the contribution of the L1 penalty and one minus the alpha value is used to weight the L2 penalty. We noticed that the model performance has increased significantly for lasso but in case of ridge there wasn’t any noticeable increase.

### 7. ELASTIC NET REGRESSION

ElasticNet Regression is the method to find the coefficients that minimize the sum of error squares by applying a penalty to these coefficients. ElasticNet combines L1 and L2 (Lasso and Ridge) approaches. As a result, it performs a more efficient smoothing process. Elastic net is a penalized linear regression model that includes both the L1(Ridge) and L2(Lasso) penalties during training.

### 8. STACKING REGRESSOR

Stacking refers to a method to blend estimators. In this strategy, some estimators are individually fitted on some training data while a final estimator is trained using the stacked predictions of these base estimators. It is sometimes tedious to find the model which will best perform on a given dataset. Stacking provides an alternative by combining the outputs of several learners, without the need to choose a model specifically. The performance of stacking is usually close to the best model and sometimes it can outperform the prediction performance of each individual model.

The stacked regressor will combine the strengths of the different regressors. However, we also see that training the stacked regressor is much more computationally expensive.

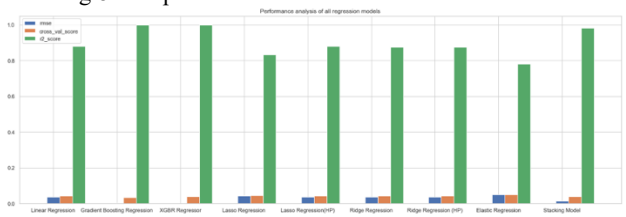
## IV. COMPARISIONS

Performance Metrics:

- Root mean squared error: Root mean squared error is the measure of how far the data points are from the regression line. Lower the root mean squared error, the better.
- Cross-validation score: Cross-validation score is used primarily to test the performance of the model when it is used to predict unseen data. Lower the cross-validation score, the better
- R-squared score: R-squared score is the measure of how close the data points are to the regression line. Higher the r-square score, the better. It lies in the range of 0 to 1.

Note: Since we are calculating the root- mean squared error on the training data itself, it is important to have a cross-validation score to make sure the model is able to perform well on non-trained data.

Fig 8: comparisions chart



| Models | rmse | cross_val_score | r2_score |
|---|---|---|---|
| Linear Regression | 0.0365 | 0.0430 | 0.8798 |
| Gradient Boosting Regression | 0.0043 | 0.0337 | 0.9983 |
| XGBR Regressor | 0.0000 | 0.0389 | 1.0000 |
| Lasso Regression | 0.0429 | 0.0440 | 0.8333 |
| Lasso Regression(HP) | 0.0365 | 0.0434 | 0.8798 |
| Ridge Regression | 0.0373 | 0.0420 | 0.8744 |
| Ridge Regression (HP) | 0.0365 | 0.0421 | 0.8744 |
| Elastic Regression | 0.0491 | 0.0494 | 0.7822 |
| Stacking Model | 0.0138 | 0.0402 | 0.9827 |

## V. LIMITATIONS & FUTURE RESEARCH

Although our dataset had very minimum number of records of 1459 rows, the machine learning models took a good amount of time for computation and to generate the results. Especially with the stacking model, the computation time was too large and if these models are fed with large datasets such as data of a state or a country, it would take really long time to generate the results. Our furture scope is to improve the performance of our model and generate results with less computational time.

## VI. CONCLUSION

Through analysis, we concluded that the stacking regression model worked best for predicting house prices. It takes into consideration all the performance metrics and we have successfully inferred that stacking model works best for this dataset. The stacking model outperforms all other models for predicting house prices because it has relatively low cross-validation and root mean squared error scores, as well as a very high r-squared score. The table below depicts the performance of the stacking model in comparison to all other models.

## VII. REFERENCES


[1] House Price Index. Federal Housing Finance Agency. https://www.fhfa.gov/ (accessed September 1, 2019).

[2] Fan C, Cui Z, Zhong X. House Prices Prediction with Machine Learning Algorithms. Proceedings of the 2018 10th International Conference on Machine Learning and Computing - ICMLC 2018. doi:10.1145/3195106.3195133.

[3] Phan TD. Housing Price Prediction Using Machine Learning Algorithms: The Case of Melbourne City, Australia. 2018 International Conference on Machine Learning and Data Engineering (ICMLDE) 2018. doi:10.1109/icmlde.2018.00017.

[4] Mu J, Wu F, Zhang A. Housing Value Forecasting Based on Machine Learning Methods. Abstract and Applied Analysis 2014;2014:1–7. doi:10.1155/2014/648047.

[5] Lu S, Li Z, Qin Z, Yang X, Goh RSM. A hybrid regression technique for house prices prediction. 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) 2017. doi:10.1109/ieem.2017.8289904.

[6] Ivanov I. vecstack. GitHub 2016. https://github.com/vecxoz/vecstack (accessed June 1, 2019). [Accessed: 01-June-2019].

[7] Wolpert DH. Stacked generalization. Neural Networks 1992;5:241–59. doi:10.1016/s0893-6080(05)80023-1.

[8] Qiu Q. Housing price in Beijing. Kaggle 2018. https://www.kaggle.com/ruiqurm/lianjia/ (accessed June 1, 2019).

[9] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. The Journal of Machine Learning Research 2011;12:2825–30